AMENDMENTS

Please amend the Specification as indicated:

On page 2, please replace the paragraph spanning lines 6-26 with the following:

> --An alternative to using separate CGI scripts to define content is a template-based HTML that actually embeds a request for the dynamic data within the HTML file itself When a specific page is requested, a pre-processor scans the file for proprietary tags that are then translated into final HTML based on the request. The final HTML is then passed back to the server and on to the browser for the user to view on their computer terminal. While the examples given have been explained in the context of HTML. Templates may be created with any Standard Generalized Markup Language (SGML) based markup language, such as Handheld Device Markup language (HDML). In fact templates can be created with any markup language or text, in fact it is not limited to SGML based languages but rather to MIME types HDML, is a markup language designed and developed by AT&T and Unwired Planet, Inc. to allow handheld devices, such as phones, access to the resources of the Internet. The specifics of the language are disclosed in "HDML Language Reference, Version 1.0," Unwired Planet, Inc., Jul. 1996. Templates with the markup in it and some scripting to execute the data and the display of the pages are separated to make generating an application a process of using an HTML template with script embedded to generate the resulting page. Examples of this technology are Active Server Pages (ASP) from Microsoft, PHP from the Apache organization or Java Server Pages (JSP) from Sun. Often these have been developed in a three-tier physical and/or logical implementation in an attempt to separate the display from the data logic. --

Please replace text following the section labeled "BRIEF DESCRIPTION OF THE DRAWINGS" spanning page 4, line 32 through page 5, line 28 with the following:

> Embodiments of the invention will now be described by way of example only, with reference to the accompanying drawings in which:

Figure 1 is a block diagram depicting a wireless network system;
Figure 2 is a block diagram depicting the major components in a system according to an embodiment of the present invention;
Figure 3 is a schematic diagram of the application element structure in a Hosted Markup Language application;
Figure 4 is a schematic representation of the form element structure in the Hosted Markup Language;
Figure 5 is a schematic representation showing the structure of the Runtime Markup Language;
Figure 6 is a block diagram of the high level components for processing of HML to generate RML;
Figure 7 is a block diagram showing the flow for the execution of components described in Figure 6;
Figure 8(a) is a schematic diagram of a Bank account query application;
Figures 8(b)-(d) is a schematic representation of an HML file for the Bank account query application;
Figure 9 shows a screen capture of a design tool, selecting forms input variables;
Figure 10 shows a screen capture of the design tool screen for selecting a list test box;
Figures 11 and 12 show screen captures of the design tool screen for specifying form data; and
Figure 13 is a screen capture of the design tool screen showing an output.

Please replace the paragraph spanning page 6, lines 1-9 with the following:

-- Referring to figure 1, a block diagram of a data communication system in which the present invention may be used is shown generally by numeral 100. The present invention is described in the context of the Internet, wherein client devices 102 make requests, through a portal or gateway 104, over the Internet 107 to web servers 108. Web servers 108 are capable of communicating via HTTP, HTTP/S or similar and providing information formatted with HTML codes the client devices 102 which may be capable of

interpreting such codes or may rely on a translation by the portal 104. In this embodiment, HTML is described as one example and the gateway may or may not translate. --

Please replace the paragraph spanning page 6, lines 11-21 with the following:

-- In fact, a gateway is not necessary for the majority of connecting devices. In a particular instance, the client devices 102 may be a cellular telephone 110 having a screen display 106, the portal 104 may be a cell phone network 114 that routes calls and data transfers from the telephone 110 to a cellular phone network. The cell phone network 114 is also capable of routing data transfers between the telephone 110 and the Internet 107. The communication between the cell phone network 114 and the Internet 107 is via the HTTP protocol, or WAP which is more likely for a phone network, the gateways typically translate the call to HTTP, which is well known in the art. Furthermore, it is assumed that the telephone 110 and the cell phone network implement the appropriate protocols for a web browser or similar that can retrieve data over the internet and translate the data file for display on the display 106. --

Please replace the paragraph spanning page 6, lines 23-29 with the following:

-- The system 100 also includes at least one host server or web server, which is a remote computer system 112 which is accessible over the internet to the cell phone network and the client devices 102. The web server 108 includes data files written in a mark up language, which may be specifically formatted for the screen display 106 of the client devices 102. This language could comprise a standard text based language similar to HTML or could include WML, HDML, or other specifically designed mark up language or simply text to send to a telephone 110. --

Please replace the paragraph spanning page 6, line 31 though page 7, line 8 with the following:

-- The web server 108 may also include an application which is run on the server and is accessible by the telephone 110 specifying a URL or similar of the application. At present, the system 100 operates by the telephone 110 transmitting a request to the cell phone network 114. The cell phone network 114 translates the request and generates a corresponding HTTP formatted message, which includes the requested URL. The HTTP request message is transmitted to the web server 108 where a data file is located. The data file must be formatted to be compatible to the display capabilities of the client devices 102. The web server calls a process, which invokes an XSL stylesheet interpreter with the appropriate HML and the stylesheet to create the formatted markup of the appropriate MIME type. In some cases both the XML input and the XSL stylesheet may be provided to the client browser to interpret if the client has an XSL built. --

Please replace the paragraph spanning page 9, lines 5-16 with the following:

-- Thus turning to figure 2, there is shown at numeral 200, the general components of a system, according to an embodiment of the present invention, for providing a unified data transfer between different devices (clients) and a server over an HTTP based network. The system 200 includes a hosted mark up language (HML) file or application 202 written in accordance with the present invention and residing on the web server 108, a plurality of style sheets 210 and a run-time program or processor 204 for processing the HML application 202 in response to an HTTP message corresponding to a request received from the client telephone 110. The HML application 202 includes a plurality of forms and pointers to external data sources. The processor 204 includes a data server 206 for retrieving data from one or more databases 216, 218 in accordance with the instructions in the HML, an XSL processor 208, and the plurality of XSL style sheets 210. --

Please replace the paragraph spanning page 9, line 23 though page 10, line 8 with the following:

-- In general, the runtime 204 is called by a client device 102 in a manner as described with reference to figure 1, which connects to the remote computer system 112 using HTTP. Based on the URL that is requested and the type of device making the request - the runtime 204 determines an appropriate form to use. The runtime 204 calls a data server component 206 to obtain data for the URL from one or more databases 218 and 216. The data server 206 retrieves the appropriate data into XML, and forwards this to the runtime which in turn adds runtime information and directory information to the data XML, the data structure that is built or populated by the runtime processor 204is termed RML, the structure of which will be described with reference to figure 5 later. The runtime processor 204calls the XSL processor 208 with the RML and an appropriate style sheet 210 for the form after the runtime 204 calls the XSL processor 208, the XSL processor generates a file that depends on how the XSL stylesheet was written. In particular a stylesheet is written for a particular MIME content-type.(notice that in the description of the RML stylesheet we have a content-type attribute) For example if it is HTML with embedded XSL instructions then the processor will generate HTML, if it is a simple test file with embedded XSL instructions then simple text will be generated. Thus if the requesting device has a specific mark-up, the runtime 204 returns the appropriate mark-up file. However, if the device does not have the specific mark-up, the run time transforms the generated WML to the appropriate markup and sends it back to the device. --

Please replace lines 4 and 5 of page 13 with the following:

-- "deviceid" stores the name of the telephone 110 and links to a value in a device lookup table 553 described later with reference to figure 7; --

Please replace the paragraph spanning page 16, lines 21-25 with the following:

-- In Step 744 the Form 623 creates a SOAP (Simple Object Access Protocol) based data server components 626 with information to create components to generate arbitrary XML data. The data server components 626 are called for each component in the application level component 309 and the form level components 11. It passes "connectionid" attribute for the data server component 626. --

Please replace the paragraph spanning page 18, lines 25-27 with the following:

-- In Step 50 the Transport component 619 returns back the generated form to the telephone 110 while setting the "contenttype" of the returned file to be consistent with device column of Table I. --

Please replace the paragraph spanning page 26, line 28 through page 27, line 11 with the following:

-- Figures 9 to 13 are screen shots of a graphical representation of the above algorithm. The use of a Field Chooser (FC) from the Select Form property page is illustrated. Note the "before" screen Figures 9 to 12 using the field chooser and the "after" screen, figure 13 of the XPath it generates. Also notice that the "form data" section of the field chooser is used. In this case it only shows elements and does not show any attributes since it is being used to get a list of elements. This enables one to do an xsl:for-each loop on a page that has a listing. The "selected element" section of the FC is showing relative XSL paths related to the chosen "selected element" in this case. In the complete view with attributes, "(text)" refers to actually getting the text of the node whereas clicking on the name of the element will generally expand to all sub-elements and attributes. This is key since an element could have text, attributes, and sub-elements. Attributes are shown with a @ in front of

the name. If an element has no attributes or sub-elements we don't display a submenu and clicking on it simply returns the text. Much of the complexity in the FC is understanding that to show schemas you need to show several views:

1) All attributers and elements
2) Only elements
3) Relative paths of elements and attributes from a selected element.--

Please replace the Abstract with the following:

-- A disclosed method for building a web-based application includes displaying a top level menu of types and showing within each level appropriate schemas. The schema information is recursively traversed to build cascading menus or toolbars. For each element, all attributes are shown, and a fully qualified path or relative path based on XSL patterns is built when a programmer selects a level. --